CNS 187 - Neural Computation
Problem Sheet 3

Handed out:	16 Oct 00
Due:	23 Oct 00, 5pm

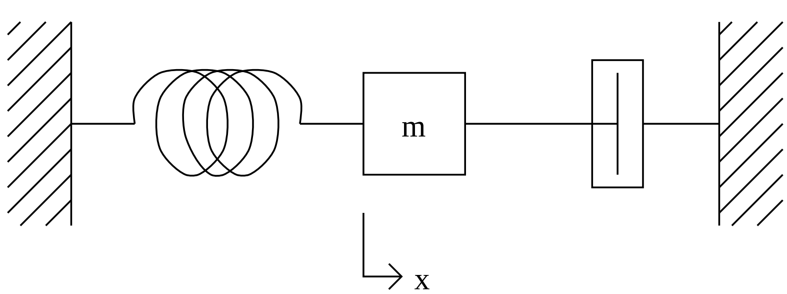
3.1 Lyapunov Energy Functions

We will consider a physical system which is a modification of the damped mass spring oscillator (see figure below).

In the original linear system, we examine a mass m attached to a linear spring (recall Hooke's law from basic physics) which experiences frictional damping proportional to velocity. The damping coefficient is denoted $\gamma > 0$, and the spring constant $\kappa > 0$. The dynamics of such a system, derived from Newton's laws, give rise to a second order linear differential equation whose behavior is easy to understand and should be quite familiar. In this problem, we introduce a nonlinear variation of the damped oscillator. To analyze it, we will use a key concept from nonlinear stability analysis: the Lyapunov, or energy function.

Consider the one-dimensional oscillator with mass m, damping coefficient γ, and a *nonlinear* spring with a characteristic $F = \kappa x(1 + \delta x^2)$; the extra spring constant is denoted $\delta > 0$. We have essentially added a cubic nonlinearity to the linear Hooke's Law.

1. Write down the dynamic equations in terms of the position x.
2. Recall your basic classical physics and write down a function describing the total energy E of the system as a function of x and its derivatives.
3. Given the dynamic equation you wrote down, find the steady-state (no change in time) of the system. Finally, noticing that the energy function you have constructed corresponds to the idea of a Lyapunov function introduced in the lectures, show that the steady state is *stable*. That is to say, show that E is bounded below, $\frac{dE}{dt} \leq 0$, and $\frac{dE}{dt} = 0$ iff $x' = x'' = 0$.



3.2 Lyapunov analysis of networks

Lyapunov analysis shows that recurrent networks with symmetric weights are guaranteed to be stable, no matter what the weights are. We also know that strictly feed-forward networks are always stable, and that there are many networks with asymmetric weights that oscillate or behave chaotically. Let's push the Lyapunov analysis to explore the middle ground.

First, let's consider a network where each neuron has its own time constant and its own activation function, with its own gain. I.e.,

$$\tau_i \dot{V}_i = -V_i + \sum_{j=1}^{N} w_{ij} S_j + I_i \tag{1}$$

$$S_i = g_i(V_i) = \alpha_i g(\beta_i V_i) \tag{2}$$

where $\tau_i > 0$ and the activation function $g(\cdot)$ is any[1] bounded monotonic increasing differentiable function (so $g'(\cdot) > 0$).

Following the original analysis, we define a function

$$\mathcal{L} = -\frac{1}{2} \sum_{i,j} w_{ij} S_i S_j + \sum_i G_i(S_i) - \sum_i S_i I_i \tag{3}$$

where $G_k(S_k) = \int_0^{S_k} g_k^{-1}(z) dz$ and $g_k^{-1}(S_k) = V_k$ is the inverse activation function for unit k.

1. Consider a symmetric network, with $w_{ij} = w_{ji}$. Give sufficient (but not necessarily necessary) conditions on α_i and β_i under which $\mathcal{L}$ is a Lyapunov function for this network. Show that $\mathcal{L}$ satisfies the three requirements for a Lyapunov function.

2. Symmetric networks with individually set activation functions (as described above) are equivalent to a subclass of networks with asymmetric weights, including feedforward networks as a limit. You will show this. Consider feedforward networks defined by

   $$\tau \dot{u}_i = -u_i + \sum_j C_{ij} g(u_j) + i_i \tag{4}$$

   where $g(\cdot)$ is the same function as above, and neurons only receive synaptic input from earlier neurons, i.e., $C_{ij} = 0$ for $i \le j$.

   For every $r > 0$, define an asymmetric recurrent network with

   $$\tau \dot{u}_i = -u_i + \sum_j C_{ij}^r g(u_j) + i_i \tag{5}$$

   where $C_{ij}^r = C_{ij}$ for $i > j$ and $C_{ij}^r = C_{ji} r^{i-j}$ for $i \le j$. Describe in words how this asymmetric network's behavior is related to the feedforward network, as r goes from 1 to ∞.

[1] For concreteness you may take $g(x) = \tanh(x)$ if you wish.

3. As a function of r, find $\tau_i, \alpha_i, \beta_i, I_i$, and $w_{ij} = w_{ji}$ such that Equation 1 is exactly equivalent to Equation 5 for $u_i = \beta_i V_i$.

4. We don't want to leave you with the impression that all networks are stable. Consider a three-node network, "ring oscillator," described by

$$\tau \dot{V}_i = -V_i + \sum_j w_{ij} g_\beta(V_i) \tag{6}$$

with $w_{13} = w_{21} = w_{32} = -1$ and other $w_{ij} = 0$, and $g_\beta(x) = g(\beta x) = \tanh(\beta x)$. First show that the network has a unique steady state ($\dot{V}_1 = \dot{V}_2 = \dot{V}_3 = 0$) at $V_1 = V_2 = V_3 = 0$. Then show that the network oscillates (i.e. that the steady state is unstable) for $\beta > \beta_0$. What is β_0?

3.3 Computing by Optimization: Graph Bipartitioning

Consider the problem of partitioning a graph with N vertices (N even) into two groups, such that the number of edges between the two groups is minimized, subject to the constraint that the number of vertices in each group is equal. We can write an energy (cost) function which can then be locally minimized by the dynamics of our neural network. A good representation is for neuron i to be on ($S_i = +1$) when vertex i is in group one, and off ($S_i = -1$) when vertex i is in group two. A good representation of an arbitrary graph is to let

$$C_{ij} = \begin{cases} 1 & \text{if there is an edge between vertex } i \text{ and vertex } j \\ 0 & \text{otherwise} \end{cases} .$$

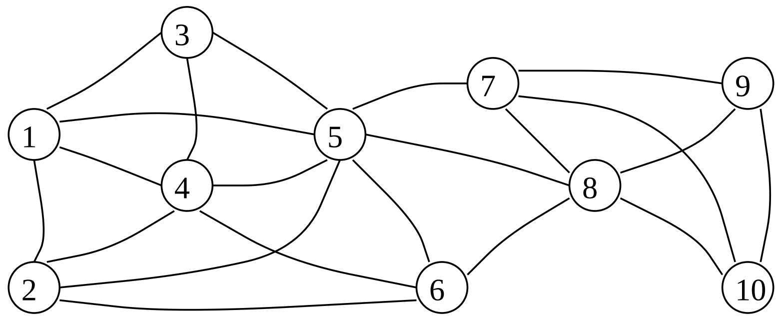
Recall that in the high gain limit our Lyapunov function is

$$\mathcal{L} = -\frac{1}{2}\left(\sum_{i,j} w_{ij} S_i S_j\right) - \left(\sum_i I_i S_i\right) ,$$

and that $\mathcal{L}$ is always reduced as the network evolves according to its dynamics.

1. Show how to construct symmetric w_{ij} and/or I_i in terms of C_{ij}, so that a global minimum of $\mathcal{L}$ gives a solution to the graph bipartitioning problem. Your formula must be *explicit*, with no free parameters (if you have a free parameter, you must specify for what range of parameters the formula will work). Prove that the global minimum of your $\mathcal{L}$ is indeed a solution to the graph bipartitioning problem. (Note: this problem is discussed in HKP 4.3, but they do not give a full proof. You should give a full proof. Consulting HKP is OK, but shouldn't be necessary.)

2. Graph Bipartitioning is an NP-complete problem. Thus, there are no known polynomial-time algorithms that are guaranteed to solve it for every graph. The natural dynamics of our network decreases $\mathcal{L}$ until it achieves a (local) minimum. Find a bipartitioning of the graph below that corresponds to a non-global local minimum of $\mathcal{L}$ (that is, changing the sign of any single $S_i \in \{-1, +1\}$ results in $\mathcal{L}$ increasing).



3. Give a near-exponential lower bound on the number of local minima in $\mathcal{L}$, for graphs of size N.

4. We have a big local minimum problem in the high-gain limit. We might hope that, in an electrical implementation of the network, we could start with low gain – where there is a unique minimum – and slowly increase the gain until, in the high-gain limit, we know that the global minimum of $\mathcal{L}$ gives a solution to the graph bipartitioning problem. Here, "slowly" means that the network is always in a steady-state, i.e., at a local minimum of its Lyapunov function for the current gain setting. Is our circuit guaranteed to find the global minimum of $\mathcal{L}$ at high gain? Discuss.

Finding optima by minimizing an "energy" function will be a common technique throughout this course, so it is important to understand both the advantages of this approach and its limitations.